

```

1 *****
2 # 입력한 문장에 대한 유튜브 영상과 해당 문장위치 바로 찾기
3 *****
4 # conda create -n youtube
5 # conda activate youtube
6 # pip install youtube-transcript-api
7 # python youtube.py
8
9 import webbrowser, urllib, re
10 from urllib import request, parse
11 from youtube_transcript_api import YouTubeTranscriptApi
12
13 *****
14 # Youtube 자막 스크립트로 부터 특정 문자열이 들어간 문장 추출
15 # 해당 동영상 스크립트 열기
16 *****
17 def find_sentence_from_Youtube_script(searchSentence, searchVideo = 'ALL') :
18     # (1) 크롬 웹 브라우저 등록.  크롬 브라우저 없는 경우 설치하고 실행
19     browser_path = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
20     webbrowser.register('chromex', None, webbrowser.BackgroundBrowser(browser_path))
21
22     # (2) 매개변수에 따라, 검색 대상 유튜브 videoid 추출하여 videoidList 리스트에 저장
23     # 찾을 문장을 키워드로 갖는 [전체 유튜브 비디오] 검색
24     if searchVideo == 'ALL' :
25         # (2) 검색시에 url은 ASCII코드만 가능, 한글 검색어가 문제가 되므로 ASCII 코드로 변경
26         # 일반적인 url에서 검색어 부분만 떼서 ASCII코드로 변경하는 루틴
27         searchUrl = 'https://www.youtube.com/results?search_query=' + searchSentence
28         parsedUrl = parse.urlparse(searchUrl)          # url을 6개 구성요소별로 분리, 튜플로 반환
29         query = parse.parse_qs(parsedUrl.query)        # query부분만 떼옴
30         parsedQuery = parse.urlencode(query, doseq=True)
31         #print("ASCII로 바꾼 QUERY ", parsedQuery)
32         searchUrl = 'https://www.youtube.com/results?' + parsedQuery
33         #print("ASCII로 바꾼 검색 URL ", searchUrl)
34
35         # (3) 찾을 문장과 관련된 Youtube 후보 리스트 생성
36         candidateHtmls = urllib.request.urlopen(searchUrl) # 해당 키워드로 Youtube 동영상들 검색
37                                                         # video ID들(11자)만 추출하여 리스트에 저장
38         videoidList = re.findall(r"watch?v=(WS{11})", candidateHtmls.read().decode())
39         print('대상 비디오 ID : ', videoidList)
40     else :
41         videoidList = [searchVideo] # 전달받은 [한 개의 videoid 만] 리스트에 저장
42
43     # (3) 찾을 비디오에 대하여 검색어가 들어 있는 위치 찾고 리스트에 저장. 브라우저에서 실행
44     allUrlPositionList = [] # 해당 문장이 들어가는 전체 영상의 위치를 갖는 url 저장용 리스트
45     noOfNoScriptVideos = 0
46     for videoid in videoidList :
47         try :
48             # 스크립트가 제공되는 영상일 경우 스크립트를 가져옴 (스크립트가 제공 안되는 영상도 있음)
49             resultScriptList = YouTubeTranscriptApi.get_transcript(videoid, languages=['ko', 'en'])
50         except :
51             resultScriptList = []
52             noOfNoScriptVideos += 1
53             print('Wn----- ID:{0} 영상은 스크립트 제공 안함 !!!Wn'.format(videoid))
54
55         # 스크립트가 제공되는 비디오만 대상으로 출력
56         if len(resultScriptList) > 0 :
57             print('*'*50)
58             print("비디오 내 스크립트 총 문장 갯수: ", len(resultScriptList))
59             print('*'*50)
60             urlString = None
61             totalFoundStatementNo = 0
62             urlPositionList = []
63             # 해당 유튜브 영상 하나에서 검색어가 들어 있는 문장만 찾아서 문장 시작위치까지 갖는 url 생성
64             for ndx, sentence in enumerate(resultScriptList) :
65                 if searchSentence in sentence['text']:
66                     totalFoundStatementNo +=1
67                     # 해당 문장 시작 시간 가져와서 초(sec)단위로 변경
68                     startTimeMS = str(sentence['start'])
69                     timeList = startTimeMS.split('.')
70                     startSecond = int(timeList[0])*60 + int(timeList[1])
71                     # 해당 영상 시작위치 url 구하기
72                     urlString = 'youtube.com/watch?v=' + videoid + '?t=' + str(startSecond)
73                     urlPositionList.append(urlString)
74                     print ('*** {0} 번째 문장 시작 위치: {0} (초)'.format(totalFoundStatementNo, startSecond))

```

```

75         print('Wt',sentence['text'], 'Wn')
76     else :
77         pass
78         #print('WtWt ', end='')
79     print ( '*** 총 [{}] 문장] 발견WnWn'.format(totalFoundStatementNo))
80     # 찾은 문장이 있으면 리스트에 추가
81     if totalFoundStatementNo > 0:
82         allUrIPositionList.append(urIPositionList)
83         # 웹브라우저로 해당 동영상 실행
84         #webbrowser.get('chromex').open(urlString, new=0)
85         pass
86
87     if len(allUrIPositionList) > 0 :
88         print('#'*50)
89         # 각 유튜브 링크에서 첫번째 문장이 나오는 위치만 출력
90         for url in allUrIPositionList:
91             print(url[0])
92
93         # 자막 제공 안하는 영상 비율 출력
94     if len(videoIdList) > 0 :
95         print('Wn총 {}개 비디오 중 {}개 ({}%)는 자막 제공 안함'.
96               format(len(videoIdList), noOfNoScriptVideos, 100*noOfNoScriptVideos/len(videoIdList)))
97
98 if __name__ == '__main__':
99     while True:
100         print('='*80)
101         print('한글 문장/단어를 입력하면 해당 문장이 들어 있는 유튜브 영상들[의 시작부분]을 찾음. Wn[q]:종료')
102         print('='*80)
103         searchSentence = input("찾을 문장: ")
104
105         if searchSentence == str('q') :
106             break
107
108         find_sentence_from_Youtube_script(searchSentence, 'ALL')
109         #find_sentence_from_Youtube_script(searchSentence, 'uEKB0afAJns')
110

```